

SOME UNDECIDABILITY RESULTS FOR ASYNCHRONOUS TRANSDUCERS AND THE BRIN-THOMPSON GROUP $2V$

JAMES BELK AND COLLIN BLEAK

ABSTRACT. Using a result of Kari and Ollinger, we prove that the torsion problem for elements of the Brin-Thompson group $2V$ is undecidable. As a result, we show that there does not exist an algorithm to determine whether an element of the rational group \mathcal{R} of Grigorchuk, Nekrashevich, and Sushchanskii has finite order. A modification of the construction gives other undecidability results about the dynamics of the action of elements of $2V$ on Cantor space. Arzhantseva, Lafont, and Minasyan proved in 2012 that there exists a finitely presented group with solvable word problem and unsolvable torsion problem. To our knowledge, $2V$ furnishes the first concrete example of such a group and gives an example of a direct undecidability result in the extended family of R. Thompson type groups.

1. INTRODUCTION

If G is a finitely presented group, the *torsion problem* for G is the problem of deciding whether a given word in the generators represents an element of finite order in G . Like the word and conjugacy problems, the torsion problem is not solvable in general [BBN59]. Perhaps more surprising is the fact that there exist finitely presented groups with solvable word problem and unsolvable torsion problem. This result was proven by Arzhantseva, Lafont, and Minasyan in 2012 [ALM12], but they did not give a specific example of such a group.

In the 1960's, Richard J. Thompson introduced a family of three groups F , T , and V , which act by homeomorphisms on an interval, a circle, and a Cantor set, respectively. These groups have a remarkable array of properties: for example, T and V were among the first known examples of finitely presented infinite simple groups. Though Thompson and McKenzie used F to construct new examples of groups with unsolvable word problem [TM00], the groups F , T , and V themselves have solvable word problem, solvable conjugacy problem, and solvable torsion problem (see [BM13]).

In 2004, Matt Brin introduced a family $\{nV\}_{n=1}^{\infty}$ of Thompson-like groups, where $1V = V$ [Br04]. Each group nV acts by piecewise-affine homeomorphisms on the direct product of n copies of the middle-thirds Cantor set. These groups are all simple [Br10] and finitely presented [Br05, HeMa12], and indeed they have type F_{∞} [KMN13, FMWZ13]. It follows from Brin's work that they all have solvable word problems. Our first main result is the following.

Received by the editors April 26, 2015.

2010 *Mathematics Subject Classification*. Primary 20F10; Secondary 20B27, 20E32, and 37B99.

Key words and phrases. Undecidable torsion problem, Brin-Thompson groups, rational group, transducer, reversible Turing machine.

Theorem 1.1. *For $n \geq 2$, the Brin-Thompson group nV has unsolvable torsion problem.*

It is easy to show that mV embeds in nV for $m \leq n$, so it suffices to prove this theorem in the case where $n = 2$. Our strategy is to use elements of $2V$ to simulate the operation of certain Turing machines. In 2008, Kari and Ollinger proved [KaOl08] that there does not exist an algorithm to determine whether a given complete, reversible Turing machine has uniformly periodic dynamics on its configuration space. We show that every such machine is topologically conjugate to an (effectively constructible) element of $2V$, and therefore there does not exist an algorithm to determine whether a given element of $2V$ has finite order.

Our second main result concerns the periodicity problem for asynchronous transducers. Roughly speaking, an *asynchronous transducer* is a finite-state automaton that converts an input string of arbitrary length to an output string. The transducer reads one symbol at a time, changing its internal state and outputting a finite sequence of symbols at each step. Asynchronous transducers are a natural generalization of *synchronous transducers*, which are required to output exactly one symbol for every symbol read.

Every transducer defines a *rational function*, which maps the space of infinite strings to itself. A transducer is *invertible* if this function is a homeomorphism. The group of all rational functions defined by invertible transducers is the *rational group* \mathcal{R} defined by Grigorchuk, Nekrashevych, and Sushchanskiĭ [GNS00]. Subgroups of \mathcal{R} are known as *automata groups*.

The idea of groups of homeomorphisms defined by transducers has a long history. Alësin [Al72] uses such a group to provide a counterexample to the unbounded Burnside conjecture. Later, Grigorchuk uses automata groups to provide a 2-group counterexample to the Burnside conjecture [Gr80] and to construct a group of intermediate growth, settling a well-known question of Milnor [Gr83]. In the last decade, the work of Bartholdi, Grigorchuk, Nekrashevich, Sidki, Štítný, and many others has advanced the theory of automata groups considerably and has brought these groups to bear on problems in geometric group theory, complex dynamics, and fractal geometry.

A transducer is *periodic* if some iterate of the corresponding rational function is equal to the identity. Our second main theorem is the following.

Theorem 1.2. *There does not exist an algorithm to determine whether a given asynchronous transducer is periodic.*

We prove this result by showing that every element of the group $2V$ is topologically conjugate to a rational function defined by a transducer. Since the torsion problem in $2V$ is undecidable, Theorem 1.2 follows.

One important problem in the theory of automata groups is the *finiteness problem*: given a finite collection of invertible transducers, is it possible to determine whether the corresponding rational homeomorphisms generate a finite group? This question was posed by Grigorchuk, Nekrashevych, and Sushchanskiĭ in [GNS00] and has since received significant attention in the literature. Gillibert [Gi13] proved that it is undecidable whether the semigroup of rational functions generated by a given collection of (not necessarily invertible) transducers is finite, which our result implies as well. Akhvai, Klimann, Lombardy, Mairesse and Picantin [AKLMP12], Klimann [Kl12], and Bondarenko, Bondarenko, Sidke, and Zapata [BBSZ13] have

also obtained partial decidability or undecidability results in various contexts. Our result settles the question for asynchronous transducers.

Theorem 1.3. *The finiteness problem for groups generated by asynchronous automata is unsolvable.*

This follows immediately from Theorem 1.2, which states that no such algorithm exists for the cyclic group generated by a single asynchronous automaton. Note that the finiteness problem is still open for groups generated by synchronous automata, which includes all Grigorchuk groups, branch groups, iterated monodromy groups, and self-similar groups.

In the last section, we show how to simulate arbitrary Turing machines using elements of $2V$, and we use the construction to prove some further undecidability results for the dynamics of elements. For example, we prove that there exists an element of $f \in 2V$ with an attracting fixed point such that the basin of the fixed point is a noncomputable set.

It is an open question whether the group $2V$ has a solvable conjugacy problem, and our result does not settle the issue. However, it does seem clear that the conjugacy problem in $2V$ must be considerably harder than in Thompson's group V . In particular, there can be no conjugacy invariant for $2V$ that gives a complete description of the dynamics, for it is not even possible to detect whether the dynamics are periodic! This contrasts sharply with Thompson's group V , in which such an invariant is easy to compute (see [BM13]).

Based on this result, it seems likely that the conjugacy problem in $2V$ is undecidable. If this is indeed the case, the group $2V$ may be useful for public-key cryptography [AAG99, KLCHKP00].

2. TURING MACHINES

In this section we define Turing machines, reversible Turing machines, and complete reversible Turing machines. Our treatment here is very similar to the one in [KaOl08], which is in turn based on the treatments in [Ku97] and [Mo96].

For the following definition, we fix two symbols L (for *left*) and R (for *right*), representing the two types of movement instructions for a Turing machine.

Definition 2.1. A *Turing machine* is an ordered triple $(\mathcal{S}, \mathcal{A}, T)$, where

- \mathcal{S} is a finite set of *states*,
- \mathcal{A} is a finite alphabet of *tape symbols*, and
- $T \subseteq (\mathcal{S} \times \{\text{L}, \text{R}\} \times \mathcal{S}) \cup (\mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{A})$ is the *transition table*.

A *tape* for a Turing machine $\mathcal{T} = (\mathcal{S}, \mathcal{A}, T)$ is any function $\tau: \mathbb{Z} \rightarrow \mathcal{A}$, i.e. any element of $\mathcal{A}^{\mathbb{Z}}$. A *configuration* of a Turing machine is a pair (s, τ) , where s is a state and τ is a tape. The set $\mathcal{S} \times \mathcal{A}^{\mathbb{Z}}$ of all configurations is called the *configuration space* for \mathcal{T} .

Each element of the transition table T is called an *instruction*. There are two types of instructions:

- (1) An instruction $(s, \delta, s') \in \mathcal{S} \times \{\text{L}, \text{R}\} \times \mathcal{S}$ is called a *move instruction*, with *initial state* s , *direction* δ , and *final state* s' .
- (2) An instruction $(s, a, s', a') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{A}$ is called a *write instruction*, with *initial state* s , *read symbol* a , *final state* s' , and *write symbol* a' .

Together, the instructions of T define a *transition relation* \rightarrow on the configuration space $\mathcal{S} \times \mathcal{A}^{\mathbb{Z}}$.

Specifically, let $W: \mathcal{A}^{\mathbb{Z}} \times \mathcal{A} \rightarrow \mathcal{A}^{\mathbb{Z}}$ and $M: \mathcal{A}^{\mathbb{Z}} \times \{\mathbf{L}, \mathbf{R}\} \rightarrow \mathcal{A}^{\mathbb{Z}}$ be the functions defined by

$$W(\tau, a)(n) = \begin{cases} a & \text{if } n = 0, \\ \tau(n) & \text{if } n \neq 0, \end{cases} \quad \text{and} \quad M(\tau, \delta)(n) = \begin{cases} \tau(n - 1) & \text{if } \delta = \mathbf{L}, \\ \tau(n + 1) & \text{if } \delta = \mathbf{R}, \end{cases}$$

for all $n \in \mathbb{Z}$. That is, W is the function that writes a symbol on the tape at position 0, while M is the function that moves the head left or right by one step. Using these functions, we can define the transition relation \rightarrow for configurations:

- (1) Each move instruction $(s, \delta, s') \in T$ specifies that

$$(s, \tau) \rightarrow (s', M(\tau, \delta))$$

for every tape $\tau \in \mathcal{A}^{\mathbb{Z}}$.

- (2) Each write instruction $(s, a, s', a') \in T$ specifies that

$$(s, \tau) \rightarrow (s', W(\tau, a'))$$

for every tape $\tau \in \mathcal{A}^{\mathbb{Z}}$ for which $\tau(0) = a$.

This completes the definition of \rightarrow , as well as all of the basic definitions for general Turing machines.

We are interested in certain kinds of Turing machines:

Definition 2.2. Let $\mathcal{T} = (\mathcal{S}, \mathcal{A}, T)$ be a Turing machine.

- (1) We say that \mathcal{T} is *deterministic* if for every configuration (s, τ) , there is at most one configuration (s', τ') so that $(s, \tau) \rightarrow (s', \tau')$.
- (2) We say that \mathcal{T} is *reversible* if \mathcal{T} is deterministic and for every configuration (s', τ') , there is at most one configuration (s, τ) such that $(s, \tau) \rightarrow (s', \tau')$.
- (3) We say that \mathcal{T} is *complete* if for every configuration (s, τ) , there is at least one configuration (s', τ') such that $(s, \tau) \rightarrow (s', \tau')$. (That is, \mathcal{T} is complete if it has no halting configurations.)

Though we have defined these conditions using the configuration space $\mathcal{S} \times \mathcal{A}^{\mathbb{Z}}$, they can be checked directly from the transition table T (see [KaOl08]).

Proposition 2.3. *If \mathcal{T} is a complete, reversible Turing machine, then the transition relation \rightarrow is a bijective function on the configuration space $\mathcal{S} \times \mathcal{A}^{\mathbb{Z}}$.*

Proof. It is clear that \rightarrow defines an injective function. The surjectivity follows from a simple counting argument on the transitions between states (see [Ku97]). \square

If \mathcal{T} is a complete, reversible Turing machine, the bijection $F: \mathcal{S} \times \mathcal{A}^{\mathbb{Z}} \rightarrow \mathcal{S} \times \mathcal{A}^{\mathbb{Z}}$ defined by the transition relation is called the *transition function* for \mathcal{T} . We say that \mathcal{T} is *uniformly periodic* if there exists an $n \in \mathbb{N}$ such that F^n is the identity function. In [KaOl08], Kari and Ollinger prove the following theorem:

Theorem 2.4 (Kari-Ollinger). *It is undecidable whether a given complete, reversible Turing machine is uniformly periodic.* \square

We shall use this theorem to prove the undecidability of the torsion problem for elements of $2V$.

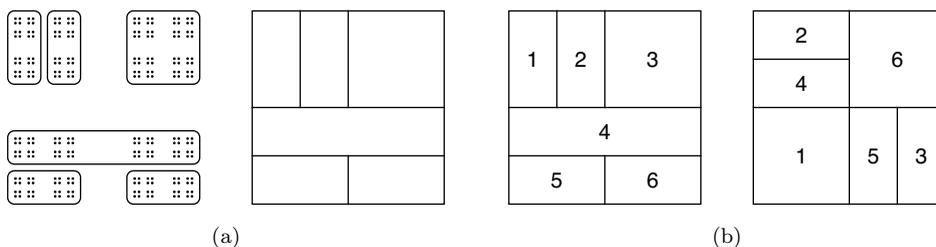


FIGURE 1. (a) A dyadic subdivision of C^2 and the corresponding pattern in the unit square. (b) A numbered pattern pair for an element of $2V$.

3. THE GROUP $2V$

In this section we define the Brin-Thompson group $2V$ and establish conventions for describing its elements. Because we wish to view the Cantor set as the infinite product space $\{0, 1\}^\infty$, our notation and terminology are slightly different from that of [Br04].

Let C be the Cantor set, which we identify with the infinite product space $\{0, 1\}^\infty$, and let $\{0, 1\}^*$ denote the set of all finite sequences of 0's and 1's. Given a finite sequence $\alpha \in \{0, 1\}^*$, the corresponding *dyadic interval* in C is the set

$$I(\alpha) = \{\alpha\omega \mid \omega \in \{0, 1\}^\infty\},$$

where $\alpha\omega$ denotes the concatenation of the finite sequence α with the infinite sequence ω . Note that the Cantor set C is itself a dyadic interval, namely, the interval $I(-)$ corresponding to the empty sequence. A *dyadic subdivision* of the Cantor set C is any partition of C into finitely many dyadic intervals.

Let C^2 denote the Cartesian product $C \times C$. A *dyadic rectangle* in C^2 is any set of the form $R(\alpha, \beta) = I(\alpha) \times I(\beta)$, where $I(\alpha)$ and $I(\beta)$ are dyadic intervals. A *dyadic subdivision* of C^2 is any partition of C^2 into finitely many dyadic rectangles. As discussed in [Br04], every dyadic subdivision of C^2 has an associated *pattern*, which is a subdivision of the unit square into rectangles. For example, Figure 1(a) shows a dyadic subdivision of C^2 and the corresponding pattern.

If $R(\alpha, \beta)$ and $R(\gamma, \delta)$ are dyadic rectangles, the *prefix replacement function* $f: R(\alpha, \beta) \rightarrow R(\gamma, \delta)$ is the function defined by

$$f(\alpha\psi, \beta\omega) = (\gamma\psi, \delta\omega),$$

for all $\psi, \omega \in \{0, 1\}^\infty$. Note that this is a bijection between $R(\alpha, \beta)$ and $R(\gamma, \delta)$.

Definition 3.1. The *Brin-Thompson group $2V$* is the group of all homeomorphisms $f: C^2 \rightarrow C^2$ with the following property: there exists a dyadic subdivision \mathcal{D} of C^2 such that f acts as a prefix replacement on each dyadic rectangle of \mathcal{D} .

Note that the images $\{f(R) \mid R \in \mathcal{D}\}$ of the rectangles of the dyadic subdivision \mathcal{D} are again a dyadic subdivision of C^2 . Since there is only one prefix replacement mapping any dyadic rectangle to any other, an element of $2V$ is entirely determined by a pair of dyadic subdivisions, together with a one-to-one correspondence between the rectangles. This lets us represent any element $f \in 2V$ by a pair of numbered patterns, as shown in Figure 1(b).

Note that the numbered pattern pair for an element $f \in 2V$ is not unique. In particular, given any numbered pattern pair for f , we can horizontally or vertically bisect a corresponding pair of rectangles in the domain and range to obtain another numbered pattern pair for f .

Nonetheless, it is possible to compute effectively using numbered pattern pairs. In [Br04], Brin describes an effective procedure to compute a numbered pattern pair for a composition fg of two elements of $2V$, given a numbered pattern pair for each element. Note that we can also effectively find a numbered pattern pair for the inverse of an element simply by switching the numbered patterns for the domain and range.

Proposition 3.2. *Given numbered pattern pairs for two elements of $2V$, there is an effective procedure to determine whether the two elements are equal.*

Proof. Let f and g be the two elements. Using Brin's procedure, we can find a numbered pattern pair for $f^{-1}g$. Then $f = g$ if and only if $f^{-1}g$ is the identity element, which occurs if and only if the numbered domain and range patterns for $f^{-1}g$ are identical. \square

It is shown in [Br05] that the group $2V$ is finitely presented, with 8 generators and 70 relations. The following proposition is implicit in [Br04] and [Br05].

Proposition 3.3. *The word problem is solvable in $2V$.*

Proof. Given any two words, we can use Brin's procedure to construct numbered pattern pairs for the corresponding elements. By Proposition 3.2, we can use these to determine whether the elements are equal. \square

We will also need the following result.

Proposition 3.4. *Given a numbered pattern pair for an element $f \in 2V$, there is an effective procedure to find a word for f .*

Proof. Given a word w , we can determine whether w represents f by computing a numbered pattern pair for w , and then comparing with f using Proposition 3.2. Therefore, we need only search through all possible words w until we find one that agrees with f . \square

See [BC10] for explicit bounds relating the word lengths of elements to the number of rectangles in a numbered pattern pair.

4. TURING MACHINES IN $2V$

The goal of this section is to prove Theorem 1.1. That is, we wish to encode any complete, reversible Turing machine as an element of $2V$ in such a way that the Turing machine is uniformly periodic if and only if the element has finite order.

Let $\mathcal{T} = (\mathcal{S}, \mathcal{A}, T)$ be a complete, reversible Turing machine. Let $\{s_1, \dots, s_m\}$ denote the states of \mathcal{T} , and let $\{I(\sigma_1), \dots, I(\sigma_m)\}$ be a corresponding dyadic subdivision of C , where $\sigma_1, \dots, \sigma_m \in \{0, 1\}^*$. Similarly let $\{a_1, \dots, a_n\}$ denote the tape symbols for \mathcal{T} , and let $\{I(\alpha_1), \dots, I(\alpha_n)\}$ be a corresponding dyadic subdivision of C , where $\alpha_1, \dots, \alpha_n \in \{0, 1\}^*$.

Given any infinite sequence $\omega \in \mathcal{A}^\infty$ of symbols, we can encode it to obtain an infinite sequence $\epsilon(\omega) \in \{0, 1\}^\infty$ of 0's and 1's as follows:

$$\epsilon(a_{i_1}, a_{i_2}, a_{i_3}, \dots) = \alpha_{i_1} \alpha_{i_2} \alpha_{i_3} \cdots;$$

that is, $\epsilon(\omega)$ is the infinite concatenation of the corresponding sequence of α 's.

Proposition 4.1. *The function $\epsilon: \mathcal{A}^\infty \rightarrow \{0, 1\}^\infty$ defined above is a bijection.*

Proof. Let $\psi_0 \in \{0, 1\}^\infty$. Since $\{I(\alpha_1), \dots, I(\alpha_n)\}$ is a dyadic subdivision of C , there exists a unique $i_1 \in \{1, \dots, n\}$ so that α_{i_1} is a prefix of ψ_0 . Then $\psi_0 = \alpha_{i_1}\psi_1$ for some $\psi_1 \in \{0, 1\}^\infty$. Then ψ_1 itself has some uniquely determined α_{i_2} as a prefix, and hence $\psi_1 = \alpha_{i_2}\psi_2$ for some $\psi_2 \in \{0, 1\}^\infty$. Continuing in this way, we can express ψ_0 uniquely as an infinite concatenation $\alpha_{i_1}\alpha_{i_2}\alpha_{i_3}\dots$. Then

$$\epsilon^{-1}(\psi_0) = (a_{i_1}, a_{i_2}, a_{i_3}, \dots),$$

which proves that ϵ is invertible. □

Now let $\Phi: \mathcal{S} \times \mathcal{A}^\mathbb{Z} \rightarrow C^2$ be the *configuration encoding* defined by

$$\Phi(s_i, \tau) = (\sigma_i \epsilon(\tau_L), \epsilon(\tau_R))$$

for every configuration (s_i, τ) , where

$$\tau_L = (\tau(-1), \tau(-2), \tau(-3), \dots) \quad \text{and} \quad \tau_R = (\tau(0), \tau(1), \tau(2), \dots).$$

That is, the first component of $\Phi(s_i, \tau)$ encodes the state s_i as well as the left half of the tape τ , while the second component of $\Phi(s_i, \tau)$ encodes the right half of the tape τ . Clearly Φ is a bijection from the configuration space $\mathcal{S} \times \mathcal{A}^\mathbb{Z}$ to C^2 .

Theorem 4.2. *Let $F: \mathcal{S} \times \mathcal{A}^\mathbb{Z} \rightarrow \mathcal{S} \times \mathcal{A}^\mathbb{Z}$ be the transition function for \mathcal{T} . Then $f_\mathcal{T} = \Phi \circ F \circ \Phi^{-1}$ is an element of $2V$.*

Proof. Since \mathcal{T} is complete and reversible, we know that F is bijective, and therefore $f_\mathcal{T}$ is bijective as well. To show that $f_\mathcal{T} \in 2V$, we need only demonstrate a dyadic subdivision \mathcal{D} of C^2 such that $f_\mathcal{T}$ acts by a prefix replacement on each rectangle of the subdivision.

The subdivision \mathcal{D} consists of the following dyadic rectangles:

- (1) For each state s_i that is the initial state of a left move instruction, \mathcal{D} includes the rectangles $\{R(\sigma_i\alpha_k, -)\}_{k=1}^n$.
- (2) For each state s_i that is the initial state of a right move instruction, \mathcal{D} includes the rectangles $\{R(\sigma_i, \alpha_k)\}_{k=1}^n$.
- (3) For each state s_i that is the initial state of write instructions, \mathcal{D} includes the rectangles $\{R(\sigma_i, \alpha_k)\}_{k=1}^n$.

Note that in each of the three cases the given rectangles are a subdivision of $R(\sigma_i, -)$. It follows that \mathcal{D} is a dyadic subdivision of C^2 . Moreover, it is easy to check that $f_\mathcal{T}$ has the right form on each rectangle of \mathcal{D} . In particular:

- (1) For a left move instruction (s_i, L, s_j) , the formula for $f_\mathcal{T}$ on each rectangle $R(\sigma_i\alpha_k, -)$ is

$$f_\mathcal{T}(\sigma_i\alpha_k\psi, \omega) = (\sigma_j\psi, \alpha_k\omega)$$

for all $\psi, \omega \in \{0, 1\}^\infty$. That is, $f_\mathcal{T}$ maps each $R(\sigma_i\alpha_k, -)$ to $R(\sigma_j, \alpha_k)$.

- (2) For a right move instruction (s_i, R, s_j) , the formula for $f_\mathcal{T}$ on each rectangle $R(\sigma_i, \alpha_k)$ is

$$f_\mathcal{T}(\sigma_i\psi, \alpha_k\omega) = (\sigma_j\alpha_k\psi, \omega)$$

for all $\psi, \omega \in \{0, 1\}^\infty$. That is, $f_\mathcal{T}$ maps each $R(\sigma_i, \alpha_k)$ to $R(\sigma_j\alpha_k, -)$.

- (3) For a write instruction (s_i, a_j, s_k, a_ℓ) , the formula for $f_\mathcal{T}$ on the rectangle $R(\sigma_i, \alpha_k)$ is

$$f_\mathcal{T}(\sigma_i\psi, \alpha_k\omega) = (\sigma_j\psi, \alpha_\ell\omega)$$

for all $\psi, \omega \in \{0, 1\}^\infty$. That is, $f_\mathcal{T}$ maps each $R(\sigma_i, \alpha_k)$ to $R(\sigma_j, \alpha_\ell)$.

We conclude that $f_\mathcal{T} \in 2V$. □

Proposition 4.3. *Given a complete, reversible Turing machine $\mathcal{T} = (\mathcal{S}, \mathcal{A}, T)$, there exists an effective procedure for choosing an element $f_{\mathcal{T}}$ as defined above and expressing it as a word in the generators of $2V$.*

Proof. Note first that it is easy to choose the dyadic subdivisions $\{I(\sigma_1), \dots, I(\sigma_m)\}$ and $\{I(\alpha_1), \dots, I(\alpha_n)\}$. For example, $(\sigma_1, \dots, \sigma_m)$ could be the m 'th term of the sequence

$$(-), (0, 1), (0, 10, 11), (0, 10, 110, 111), (0, 10, 110, 1110, 1111), \dots,$$

and $(\alpha_1, \dots, \alpha_n)$ could be the n 'th term of this sequence.

Once the subdivisions are chosen, we can use the formulas given in the proof of Theorem 4.2 to construct a numbered pattern pair for the element $f_{\mathcal{T}}$. Finally, we can use Proposition 3.4 to compute a word for $f_{\mathcal{T}}$. □

Proposition 4.4. *The element $f_{\mathcal{T}}$ has finite order if and only if \mathcal{T} is uniformly periodic.*

Proof. Since $f_{\mathcal{T}} = \Phi \circ F \circ \Phi^{-1}$, it follows that $(f_{\mathcal{T}})^p = \Phi \circ F^p \circ \Phi^{-1}$ for each p , so $(f_{\mathcal{T}})^p$ is the identity if and only if F^p is the identity function. □

This completes the proof of Theorem 1.1. Combining this with the result of Kari and Ollinger (Theorem 2.4 above), we conclude that the torsion problem in $2V$ is undecidable. The following theorem may shed some light on the nature of this result:

Theorem 4.5. *For each $n \in \mathbb{N}$, let $\Omega(n)$ be the maximum possible order of a torsion element of $2V$ having at most n rectangles in its numbered pattern pair. Then Ω is not bounded above by any computable function $\mathbb{N} \rightarrow \mathbb{N}$.*

Proof. Suppose to the contrary that Ω were bounded above by a computable function $\gamma: \mathbb{N} \rightarrow \mathbb{N}$. Then, given any element $f \in 2V$ with n rectangles in its numbered pattern pair, it would be a simple matter to determine whether f has finite order. Specifically, we could first compute $\gamma(n)$, then compute the powers $f, f^2, f^3, \dots, f^{\gamma(n)}$, and finally check to see if any one of these is the identity. Since the torsion problem in $2V$ is undecidable, it follows that no such function γ exists. □

Thus the function $\Omega(n)$ must grow very quickly, e.g. on the order of the busy beaver function (see [Ra62]).

We end this section with an example that illustrates the construction of $f_{\mathcal{T}}$.

Example 4.6. Consider the Turing machine \mathcal{T} , with four states $\{s_1, s_2, s_3, s_4\}$ and three symbols $\{a_1, a_2, a_3\}$, which obeys the following rules:

- (1) From state s_1 , move the head right and go to state s_2 .
- (2) From state s_2 , read the input symbol $\tau(0)$:
 - (a) If $\tau(0) = a_1$, then write a_1 and go back to state s_1 .
 - (b) If $\tau(0) = a_2$, then write a_3 and go back to state s_1 .
 - (c) If $\tau(0) = a_3$, then write a_2 and go to state s_3 .
- (3) From state s_3 , move the head left and go to state s_4 .
- (4) From state s_4 , read the input symbol $\tau(0)$:
 - (a) If $\tau(0) = a_1$, then write a_1 and go back to state s_3 .
 - (b) If $\tau(0) = a_2$, then write a_2 and go to state s_1 .
 - (c) If $\tau(0) = a_3$, then write a_3 and go back to state s_3 .

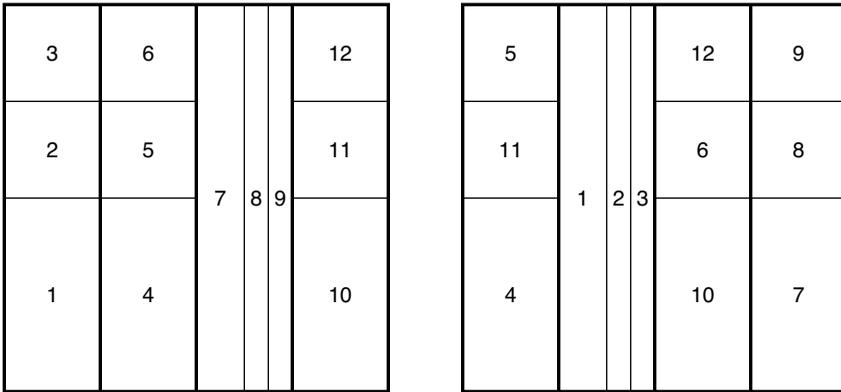


FIGURE 2. The element of $2V$ corresponding to the Turing machine in Example 4.6. The four main vertical rectangles correspond to the four states $\{s_1, s_2, s_3, s_4\}$.

That is, the transition table for \mathcal{T} is the set

$$\{(s_1, R, s_2), (s_2, a_1, s_1, a_1), (s_2, a_2, s_1, a_3), (s_2, a_3, s_3, a_2), \\ (s_3, L, s_4), (s_4, a_1, s_3, a_1), (s_4, a_2, s_1, a_2), (s_4, a_3, s_3, a_3)\}.$$

It is easy to check that \mathcal{T} is complete and reversible. To make a corresponding element of $2V$, let $(\sigma_1, \sigma_2, \sigma_3, \sigma_4) = (00, 01, 10, 11)$, and let $(\alpha_1, \alpha_2, \alpha_3) = (0, 10, 11)$. Then the resulting element $f_{\mathcal{T}} \in 2V$ is shown in Figure 2.

5. TRANSDUCERS

In this section we show that it is undecidable whether the rational function defined by a given asynchronous transducer has finite order (Theorem 1.2). This settles the finiteness problem for asynchronous automata groups (Theorem 1.3).

We begin by briefly reviewing the relevant facts about transducers. See [Gl63] for a thorough introduction to transducers, and [GNS00] for a discussion of transducers in the context of group theory.

Definition 5.1. An asynchronous *transducer* is an ordered quadruple $(\mathcal{A}, \mathcal{S}, s_0, \tau)$, where

- \mathcal{A} is a finite alphabet,
- \mathcal{S} is a finite set of states,
- $s_0 \in \mathcal{S}$ is the *initial state*, and
- $\tau: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \times \mathcal{A}^*$ is the *transition function*, where \mathcal{A}^* denotes the set of all finite strings over \mathcal{A} .

Given a transducer $(\mathcal{A}, \mathcal{S}, s_0, \tau)$ and an infinite *input string* $\alpha_1\alpha_2\alpha_3 \dots \in \mathcal{A}^\infty$, the corresponding *state sequence* $\{s_i\}$ and *output sequence* $\{\beta_i\}$ are defined recursively by

$$(s_i, \beta_i) = \tau(s_{i-1}, \alpha_i).$$

The concatenation $\beta_1\beta_2 \dots$ of the output sequence is called the *output string*. This string is usually infinite, but will be finite if only finitely many β_i 's are nonempty.

We say that the transducer is *nondegenerate* if every infinite input string results in an infinite output string. In this case, the function $f: \mathcal{A}^\infty \rightarrow \mathcal{A}^\infty$ mapping each input string to the corresponding output string is called the *rational function* defined by the given transducer. Rational functions are always continuous, and any invertible rational function is a homeomorphism.

For a given finite alphabet \mathcal{A} , the *rational group* \mathcal{R} is the group consisting of all invertible rational homeomorphisms of \mathcal{A}^∞ . It is proven in [GNS00] that \mathcal{R} forms a group and that the isomorphism type of \mathcal{R} does not depend on the size of \mathcal{A} , as long as \mathcal{A} has at least two letters.

Our goal in this section is to prove the following theorem.

Theorem 5.2. *The rational group \mathcal{R} has a subgroup isomorphic to $2V$.*

Since $2V$ has unsolvable torsion problem, Theorems 1.2 and 1.3 will follow immediately.

We begin with the following proposition, which will help us to combine rational functions together.

Proposition 5.3. *Let \mathcal{A} be a finite alphabet, let $\{\alpha_1, \dots, \alpha_n\}$ be a complete prefix code over \mathcal{A} , and let $f_1, \dots, f_n: \mathcal{A}^* \rightarrow \mathcal{A}^*$ be rational functions. Define a function $f: \mathcal{A}^* \rightarrow \mathcal{A}^*$ by*

$$f(\alpha_i\omega) = f_i(\omega)$$

for all $i \in \{1, \dots, n\}$ and $\omega \in \mathcal{A}^*$. Then f is a rational function.

Proof. Let $\{\beta_1, \dots, \beta_m\}$ be the set of all proper prefixes of strings in $\{\alpha_1, \dots, \alpha_n\}$, where β_1 is the empty string. For each $i \in \{1, \dots, n\}$, let $(\mathcal{A}, \mathcal{S}_i, s_{0i}, \tau_i)$ be a transducer for f_i . Define a transducer $(\mathcal{A}, \mathcal{S}, s_0, \tau)$ as follows:

- The state set \mathcal{S} is the disjoint union $\{\beta_1, \dots, \beta_m\} \uplus \mathcal{S}_1 \uplus \dots \uplus \mathcal{S}_n$.
- The initial state s_0 is the empty string β_1 .
- The transition function $\tau: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined by

$$\tau(s, a) = \begin{cases} (\beta_j, -) & \text{if } s = \beta_i \text{ and } \beta_i a = \beta_j, \\ (s_{0j}, -) & \text{if } s = \beta_i \text{ and } \beta_i a = \alpha_j, \\ \tau_i(s, a) & \text{if } s \in \mathcal{S}_i. \end{cases}$$

It is easy to check that the rational function defined by this transducer is the desired function f . □

Now consider the four-element alphabet $\mathcal{A} = \{00, 01, 10, 11\}$. Let $\pi: \mathcal{A}^\infty \rightarrow C^2$ be the function defined by

$$\pi(\epsilon_1\delta_1, \epsilon_2\delta_2, \epsilon_3\delta_3, \dots) = (\epsilon_1\epsilon_2\epsilon_3 \dots, \delta_1\delta_2\delta_3 \dots).$$

Given any function $f: C^2 \rightarrow C^2$, let $f^\pi = \pi^{-1} \circ f \circ \pi$ denote the corresponding function $\mathcal{A}^\infty \rightarrow \mathcal{A}^\infty$. We shall prove that the mapping $f \mapsto f^\pi$ defines a monomorphism from $2V$ to \mathcal{R} .

Proposition 5.4. *Let $\alpha, \beta \in \{0, 1\}^*$, and let $\mu_{\alpha, \beta}: C^2 \rightarrow C^2$ be the function*

$$\mu_{\alpha, \beta}(\psi, \omega) = (\alpha\psi, \beta\omega).$$

Then $\mu_{\alpha, \beta}^\pi$ is a rational function.

Proof. Suppose that $\alpha \in \{0, 1\}^m$ has length m and β has length n . Consider the transducer $(\mathcal{A}, \mathcal{S}, s_0, \tau)$ defined as follows:

- The alphabet \mathcal{A} is $\{00, 01, 10, 11\}$.
- The state set \mathcal{S} is $\{0, 1\}^m \times \{0, 1\}^n$.
- The initial state s_0 is (α, β) .
- The transition function $\tau: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \times \mathcal{A}^*$ is defined by

$$\tau((\delta_1 \cdots \delta_m, \epsilon_1 \cdots \epsilon_n), \delta_{m+1}\epsilon_{n+1}) = ((\delta_2 \cdots \delta_{m+1}, \epsilon_2 \cdots \epsilon_{n+1}), \delta_1\epsilon_1).$$

It is easy to check that the rational function defined by this transducer is $\mu_{\alpha, \beta}^\pi$. \square

Proposition 5.5. *If $f \in 2V$, then f^π is a rational function.*

Proof. Let $\{R(\alpha_1, \beta_1), \dots, R(\alpha_n, \beta_n)\}$ be a dyadic partition of C^2 so that f is linear on each $R(\alpha_i, \beta_i)$. By subdividing if necessary, we may assume that all of the strings $\alpha_1, \dots, \alpha_n$ and β_1, \dots, β_n have the same length. Let $R(\gamma_i, \delta_i) = f(R(\alpha_i, \beta_i))$ for each i , and let ϵ_i be the common initial prefix of $\pi(R(\alpha_i, \beta_i))$. Then f^π is given by the formula

$$f^\pi(\epsilon_i\omega) = \mu_{\gamma_i, \delta_i}^\pi(\omega)$$

for each $i \in \{1, \dots, n\}$ and each $\omega \in \{00, 01, 10, 11\}^*$. By Proposition 5.4, each of the functions $\mu_{\gamma_i, \delta_i}^\pi$ is rational. By Proposition 5.3, it follows that f^π is rational as well. \square

This completes the proof of Theorem 5.2, and hence of Theorems 1.2 and 1.3.

6. ALLOWING HALTING

In this section, we briefly discuss how to simulate incomplete Turing machines using elements of $2V$, and we sketch the proofs of some further undecidability results. Similar results for general piecewise-affine functions can be found in [BBKPT01].

Definition 6.1. Let \mathcal{T} be a reversible Turing machine, and let (s, τ) be a configuration for \mathcal{T} .

- (1) We say that (s, τ) is a *halting configuration* if there does not exist any configuration (s', τ') such that $(s, \tau) \rightarrow (s', \tau')$.
- (2) We say that (s, τ) is an *inverse halting configuration* if there does not exist any configuration (s', τ') such that $(s', \tau') \rightarrow (s, \tau)$.

A Turing machine that reaches a halting configuration is said to *halt*.

If \mathcal{T} is incomplete, then the transition function $F: \mathcal{S} \times \mathcal{A}^\mathbb{Z} \rightarrow \mathcal{S} \times \mathcal{A}^\mathbb{Z}$ on the configuration space is only partially defined and is injective but not surjective. Specifically, if \mathcal{H} is the set of halting configurations and $\overline{\mathcal{H}}$ is the set of inverse halting configurations, then F restricts to a bijection $\mathcal{H}^c \rightarrow \overline{\mathcal{H}}^c$, where \mathcal{H}^c and $\overline{\mathcal{H}}^c$ denote the complements of \mathcal{H} and $\overline{\mathcal{H}}$ in the configuration space.

Our construction of the corresponding element $f_\mathcal{T} \in 2V$ is only a slight modification of the construction from Section 4. To start, we subdivide C^2 into three dyadic rectangles:

$$R_\zeta = R(0, 0), \quad R_\mathcal{T} = R(-, 1), \quad R_\eta = R(1, 0).$$

We will use R_η and R_ζ for halting and inverse halting, respectively, while $R_\mathcal{T}$ will be used for configurations of the Turing machine.

Now, let $\{s_1, \dots, s_m\}$ denote the states of \mathcal{T} , and let $\{I(\sigma_1), \dots, I(\sigma_m)\}$ be a corresponding subdivision of C , where $\sigma_1, \dots, \sigma_m \in \{0, 1\}^*$. Similarly, let $\{a_1, \dots, a_n\}$

denote the tape symbols for \mathcal{T} , and let $\{I(\alpha_1), \dots, I(\alpha_n)\}$ be a corresponding dyadic subdivision of C , where $\alpha_1, \dots, \alpha_n \in \{0, 1\}^*$.

Let $\epsilon: \mathcal{A}^\infty \rightarrow \{0, 1\}^\infty$ be an encoding function derived from $(\alpha_1, \dots, \alpha_n)$, and let $\Phi: \mathcal{S} \times \mathcal{A}^\mathbb{Z} \rightarrow C^2$ be the configuration encoding defined by

$$\Phi(s_i, \tau) = (\sigma_i \epsilon(\tau_L), 1 \epsilon(\tau_R)).$$

Note that Φ is no longer surjective—its image is the rectangle $R_{\mathcal{T}}$. Moreover, note that $\Phi(\mathcal{H})$ and $\Phi(\overline{\mathcal{H}})$ are each the union of finitely many dyadic rectangles.

Let $f_{\mathcal{T}}$ be any element of $2V$ that satisfies the following conditions:

- (1) $f_{\mathcal{T}}$ agrees with $\Phi \circ F \circ \Phi^{-1}$ on $\Phi(\mathcal{H}^c)$.
- (2) $f_{\mathcal{T}}$ maps R_ζ bijectively onto $R_\zeta \cup \Phi(\overline{\mathcal{H}})$.
- (3) $f_{\mathcal{T}}$ maps $\Phi(\mathcal{H}) \cup R_\eta$ bijectively onto R_η .

Such an element can be constructed effectively. For example, to construct the portion of $f_{\mathcal{T}}$ on R_ζ , we need only enumerate the dyadic rectangles R_1, \dots, R_k of $\Phi(\overline{\mathcal{H}})$, then choose a dyadic subdivision \mathcal{D} of R_ζ into $k + 1$ rectangles, and finally choose a one-to-one correspondence between the rectangles of \mathcal{D} and $\{R_1, \dots, R_k, R_\zeta\}$.

The element $f_{\mathcal{T}}$ constructed above simulates the Turing machine \mathcal{T} , in the sense that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{H}^c & \xrightarrow{F} & \overline{\mathcal{H}}^c \\ \Phi \downarrow & & \downarrow \Phi \\ \Phi(\mathcal{H}^c) & \xrightarrow{f_{\mathcal{T}}} & \Phi(\overline{\mathcal{H}}^c) \end{array}$$

Whenever a configuration halts, $f_{\mathcal{T}}$ maps the corresponding point to R_η . Similarly, $f_{\mathcal{T}}$ maps points from R_ζ onto inverse halting configurations to “fill in” the bijection.

Example 6.2. Consider the incomplete Turing machine \mathcal{T} with two states $\{s_1, s_2\}$ and two symbols $\{a_1, a_2\}$, which obeys the following rules:

- (1) From state s_1 , move the head right and go to state s_2 .
- (2) From state s_2 , read the input symbol $\tau(0)$:
 - (a) If $\tau(0) = a_1$, then write a_2 and go back to state s_1 .
 - (b) If $\tau(0) = a_2$, then halt.

That is, the transition table for \mathcal{T} is the set

$$\{(s_1, R, s_2), (s_2, a_1, s_1, a_2)\}.$$

It is easy to check that \mathcal{T} is reversible. The halting configurations for \mathcal{T} are

$$\mathcal{H} = \{(s_2, \tau) \mid \tau \in \mathcal{A}^\mathbb{Z} \text{ and } \tau(0) = a_2\},$$

and the inverse halting configurations are

$$\overline{\mathcal{H}} = \{(s_1, \tau) \mid \tau \in \mathcal{A}^\mathbb{Z} \text{ and } \tau(0) = a_1\}.$$

To make a corresponding element of $2V$, let $(\sigma_1, \sigma_2) = (0, 1)$ and $(\alpha_1, \alpha_2) = (0, 1)$. Then one possible choice for $f_{\mathcal{T}}$ is shown in Figure 3.

For our purposes, incomplete Turing machines are useful primarily because of their relation to the halting problem. Before we can discuss this, we must restrict ourselves to a class of configurations that can be specified with a finite amount of information. First, we fix a *blank symbol* a_1 from \mathcal{A} , and we define a configuration

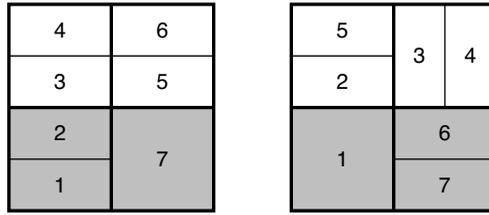


FIGURE 3. The element of $2V$ corresponding to the Turing machine in Example 6.2. The subrectangles R_ζ and R_η are shown in gray, while $R_\mathcal{T}$ is shown in white.

(s_i, τ) to be *finite* if the tape τ has the blank symbol in all but finitely many locations. Then any finite configuration (s_i, τ) can be specified using only the state s_i and some finite subsequence $(\tau(-n), \dots, \tau(n))$ of the tape whose complement consists entirely of blank symbols.

Theorem 6.3. *Any Turing machine can be effectively simulated by a reversible Turing machine.*

Proof. This was proven in [Be73] for a 3-tape reversible Turing machine, and improved to a 1-tape, 2-symbol machine in [MSG89]. □

Corollary 6.4. *It is not decidable, given a reversible Turing machine and a finite starting configuration, whether the machine will halt.*

Proof. This follows immediately from Theorem 6.3 and Turing’s theorem on the unsolvability of the halting problem for general Turing machines. □

We wish to interpret this result in the context of $2V$. We begin by defining points in C^2 that correspond to finite configurations.

Definition 6.5. A point $(\psi, \omega) \in C^2$ is *dyadic* if ψ and ω have only finitely many 1’s.

Note that a dyadic point in C^2 can be specified with a finite amount of information, namely, the initial nonzero subsequences of ψ and ω . Assuming the sequence α_1 corresponding to the blank symbol a_1 is a string of finitely many 0’s, the dyadic points in $R_\mathcal{T}$ are precisely the points that correspond to finite configurations of \mathcal{T} .

Theorem 6.6. *It is not decidable, given an element $f \in 2V$, a dyadic point $p \in C^2$, and a dyadic rectangle $R \subseteq C^2$, whether the orbit of p under f contains a point in R .*

Proof. Let \mathcal{T} be a reversible Turing machine, let (s_i, τ) be a starting configuration for \mathcal{T} , and let $f_\mathcal{T}$ be the element constructed above. Then \mathcal{T} eventually halts starting at (s_i, τ) if and only if the point $\Phi(s_i, \tau)$ eventually maps into the rectangle R_η under $f_\mathcal{T}$. By Corollary 6.4, we cannot decide whether \mathcal{T} will halt, so the given problem must be undecidable as well. □

Theorem 6.7. *It is not decidable, given an element $f \in 2V$ and two dyadic points $p, q \in C^2$, whether the orbit of p under f converges to q .*

Proof. Let \mathcal{T} be a reversible Turing machine, and let (s_i, τ) be a starting configuration for \mathcal{T} . Let f_τ be the element constructed above, with the function f_τ chosen so that $f_\tau(1\psi, 0\omega) = f_\tau(10\psi, 00\omega)$ for $(1\psi, 0\omega) \in R_\eta$. Note then that the orbit of every point in R_η converges to $(1\bar{0}, \bar{0})$. Therefore, \mathcal{T} eventually halts starting at (s_i, τ) if and only if the orbit of $\Phi(s_i, \tau)$ converges to $(1\bar{0}, \bar{0})$. By Corollary 6.4, we cannot decide whether \mathcal{T} will halt, so the given problem must be undecidable as well. \square

Theorem 6.8. *There exists an element $f \in 2V$ with an attracting dyadic fixed point $p \in C^2$ such that $B(p) \cap D$ is a noncomputable set, where $B(p)$ is the basin of attraction of p and D is the set of dyadic points in C^2 .*

Proof. It follows from Theorem 6.3 that there exists a reversible Turing machine \mathcal{T} that is computation universal, meaning that it can be used to simulate the operation any other Turing machine. Such a machine has the property that, given a starting configuration (s_i, τ) , there does not exist an algorithm to determine whether \mathcal{T} halts. (That is, the halting problem is undecidable in the context of this one machine.) If we use this machine to construct an element $f_\tau \in 2V$ as in the proof of Theorem 6.7, then f_τ will have the desired property. \square

ACKNOWLEDGMENTS

The authors would like to thank Matthew Brin, Rostislav Grigorchuk, Conchita Martinez-Perez, Francesco Matucci, Volodia Nekrashevich, and Brita Nucinkis for helpful conversations.

The second author wishes to acknowledge partial support by EPSRC grant EP/H011978/1 during a period when he was conducting some of the research that led to the creation of this article.

REFERENCES

- [AKLMP12] Ali Akhavi, Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Matthieu Picantin, *On the finiteness problem for automaton (semi)groups*, Internat. J. Algebra Comput. **22** (2012), no. 6, 1250052, 26, DOI 10.1142/S021819671250052X. MR2974106
- [Al72] S. V. Alešin, *Finite automata and the Burnside problem for periodic groups* (Russian), Mat. Zametki **11** (1972), 319–328. MR0301107
- [AAG99] Iris Anshel, Michael Anshel, and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett. **6** (1999), no. 3-4, 287–291, DOI 10.4310/MRL.1999.v6.n3.a3. MR1713130
- [ALM12] Goulnara Arzhantseva, Jean-François Lafont, and Ashot Minasyan, *Isomorphism versus commensurability for a class of finitely presented groups*, J. Group Theory **17** (2014), no. 2, 361–378, DOI 10.1515/jgt-2013-0050. MR3176665
- [BM13] James Belk and Francesco Matucci, *Conjugacy and dynamics in Thompson’s groups*, Geom. Dedicata **169** (2014), 239–261, DOI 10.1007/s10711-013-9853-2. MR3175247
- [BBN59] Gilbert Baumslag, W. W. Boone, and B. H. Neumann, *Some unsolvable problems about elements and subgroups of groups*, Math. Scand. **7** (1959), 191–201. MR0163948
- [Be73] C. H. Bennett, *Logical reversibility of computation*, IBM J. Res. Develop. **17** (1973), 525–532. MR0449020
- [BBKPT01] Vincent D. Blondel, Olivier Bournez, Pascal Koiran, Christos H. Papadimitriou, and John N. Tsitsiklis, *Deciding stability and mortality of piecewise affine dynamical systems*, Theoret. Comput. Sci. **255** (2001), no. 1-2, 687–696, DOI 10.1016/S0304-3975(00)00399-6. MR1819103

- [BBSZ13] Ievgen V. Bondarenko, Natalia V. Bondarenko, Said N. Sidki, and Flavia R. Zapata, *On the conjugacy problem for finite-state automorphisms of regular rooted trees*, Groups Geom. Dyn. **7** (2013), no. 2, 323–355, DOI 10.4171/GGD/184. With an appendix by Raphaël M. Jungers. MR3054572
- [Br04] Matthew G. Brin, *Higher dimensional Thompson groups*, Geom. Dedicata **108** (2004), 163–192, DOI 10.1007/s10711-004-8122-9. MR2112673
- [Br05] Matthew G. Brin, *Presentations of higher dimensional Thompson groups*, J. Algebra **284** (2005), no. 2, 520–558, DOI 10.1016/j.jalgebra.2004.10.028. MR2114568
- [Br10] Matthew G. Brin, *On the baker’s map and the simplicity of the higher dimensional Thompson groups nV* , Publ. Mat. **54** (2010), no. 2, 433–439, DOI 10.5565/PUBLMAT_54210_07. MR2675931
- [BC10] José Burillo and Sean Cleary, *Metric properties of higher-dimensional Thompson’s groups*, Pacific J. Math. **248** (2010), no. 1, 49–62, DOI 10.2140/pjm.2010.248.49. MR2734164
- [FMWZ13] Martin G. Fluch, Marco Marschler, Stefan Witzel, and Matthew C. B. Zaremsky, *The Brin-Thompson groups sV are of type F_∞* , Pacific J. Math. **266** (2013), no. 2, 283–295, DOI 10.2140/pjm.2013.266.283. MR3130623
- [Gi13] Pierre Gillibert, *The finiteness problem for automaton semigroups is undecidable*, Internat. J. Algebra Comput. **24** (2014), no. 1, 1–9, DOI 10.1142/S0218196714500015. MR3189662
- [Gl63] V. M. Gluškov, *The abstract theory of automata. I* (Hungarian), Magyar Tud. Akad. Mat. Fiz. Oszt. Közl. **13** (1963), 287–309. MR0158807
- [Gr80] R. I. Grigorčuk, *On Burnside’s problem on periodic groups* (Russian), Funktsional. Anal. i Prilozhen. **14** (1980), no. 1, 53–54. MR565099
- [Gr83] R. I. Grigorčuk, *On the Milnor problem of group growth* (Russian), Dokl. Akad. Nauk SSSR **271** (1983), no. 1, 30–33. MR712546
- [GNS00] R. I. Grigorčuk, V. V. Nekrashevich, and V. I. Sushchanskii, *Automata, dynamical systems, and groups* (Russian, with Russian summary), Tr. Mat. Inst. Steklova **231** (2000), no. Din. Sist., Avtom. i Beskon. Gruppy, 134–214; English transl., Proc. Steklov Inst. Math. **4** (2000), 128–203. MR1841755
- [HeMa12] Johanna Hennig and Francesco Matucci, *Presentations for the higher-dimensional Thompson groups nV* , Pacific J. Math. **257** (2012), no. 1, 53–74, DOI 10.2140/pjm.2012.257.53. MR2948458
- [KaOl08] Jarkko Kari and Nicolas Ollinger, *Periodicity and immortality in reversible computing*, Mathematical foundations of computer science 2008, Lecture Notes in Comput. Sci., vol. 5162, Springer, Berlin, 2008, pp. 419–430, DOI 10.1007/978-3-540-85238-4_34. MR2539389
- [Kl12] Ines Klimann, *The finiteness of a group generated by a 2-letter invertible-reversible Mealy automaton is decidable*, 30th International Symposium on Theoretical Aspects of Computer Science, LIPIcs. Leibniz Int. Proc. Inform., vol. 20, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2013, pp. 502–513. MR3090008
- [KLCHKP00] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park, *New public-key cryptosystem using braid groups*, Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA), Lecture Notes in Comput. Sci., vol. 1880, Springer, Berlin, 2000, pp. 166–183, DOI 10.1007/3-540-44598-6_10. MR1850042
- [KMN13] Dessislava H. Kochloukova, Conchita Martínez-Pérez, and Brita E. A. Nucinkis, *Cohomological finiteness properties of the Brin-Thompson-Higman groups $2V$ and $3V$* , Proc. Edinb. Math. Soc. (2) **56** (2013), no. 3, 777–804, DOI 10.1017/S0013091513000369. MR3109758
- [Ku97] Petr Kůrka, *On topological dynamics of Turing machines*, Theoret. Comput. Sci. **174** (1997), no. 1-2, 203–216, DOI 10.1016/S0304-3975(96)00025-4. MR1439236
- [TM00] Ralph McKenzie and Richard J. Thompson, *An elementary construction of unsolvable word problems in group theory*, Word problems: decision problems and the Burnside problem in group theory (Conf., Univ. California, Irvine, Calif. 1969; dedicated to Hanna Neumann), Studies in Logic and the Foundations of Math., vol. 71, North-Holland, Amsterdam, 1973, pp. 457–478. MR0396769

- [Mo96] Kenichi Morita, *Universality of a reversible two-counter machine*, Universal machines and computations (Paris, 1995), Theoret. Comput. Sci. **168** (1996), no. 2, 303–320, DOI 10.1016/S0304-3975(96)00081-3. MR1422960
- [MSG89] Kenichi Morita, A. Shirasaki, and Y. Gono. *A 1-tape 2-symbol reversible Turing machine*. Trans. IEICE **72** (1989), 223–228.
- [Ra62] T. Radó, *On non-computable functions*, Bell System Tech. J. **41** (1962), 877–884. MR0133229

MATHEMATICS PROGRAM, BARD COLLEGE, P.O. BOX 5000, ANNANDALE, NEW YORK 12504

E-mail address: `belk@bard.edu`

SCHOOL OF MATHEMATICS AND STATISTICS, MATHEMATICAL INSTITUTE, UNIVERSITY OF ST ANDREWS, NORTH HAUGH, ST ANDREWS, FIFE KY16 9SS, SCOTLAND

E-mail address: `cb211@st-andrews.ac.uk`